

TEST PLAN

<https://the-internet.herokuapp.com/login>

Submitted By:

Vinita Kumawat(fs40_393032)

&

Palak Mangal(fs40_360629)

Date:10/04/2025 to 12/04/2025

Index Table

S. No.	Contents	Page No.
1	Introduction	
2	Objectives	
3	Scope	
4	Testable Features	
5	Testing Approach	
6	Role/Responsibility	
7	Test Schedule	
8	Test Deliverables	
9	Entry and Exit Criteria	
10	Tools	
11	Risks & Mitigation	
12	Approvals	

➤ Introduction

This test plan outlines the strategy for validating the login functionality of the website <https://theinternet.herokuapp.com/login>. It includes both manual and automated testing approaches. The automation will follow the **Page Object Model (POM)** and **Cucumber BDD** framework using Selenium and Java.

➤ Objectives

- Ensure the login functionality works as intended for valid and invalid user credentials.
- Validate UI elements, error messages, and field validations.
- Automate the login test scenarios using a maintainable POM structure with Cucumber for BDD-style test cases.
- Deliver clean and comprehensive test reports.

➤ Scope

In-Scope:

- Functional testing of login page manually and through automation.
- Positive and negative login scenarios.
- Field validation and error message testing.
- UI validation (input field behaviour, password masking, etc.).
- Reporting using Extent/Allure.

Out of Scope:

- Performance testing

➤ Testable Features

- Username and password input fields.
- Login button functionality.
- Error message display for invalid or empty input.
- Successful login flow and navigation to secure area.
- Logout option post-login.
- Field validations (blank, incorrect inputs).

➤ Testing Approach

Manual Testing

- Prepare test cases in spreadsheet format.
- Execute each test case and record actual results.
- Validate UI behaviour and user flow.
- Log and track bugs using Jira.

Automation Testing

- Use Selenium with Java.
- Follow Page Object Model (POM) structure for reusable page components.
- Use Cucumber for writing human-readable feature files.
- Map steps to definitions and implement actions via page classes.
- Generate reports using Extent/Allure.
- Execute test scripts on multiple browsers.

➤ Roles and Responsibilities

Role	Responsibility
Vinita	Write and execute manual test cases, automate scenarios, report bugs
Palak	Implement POM, write Cucumber features, generate reports

➤ Test Schedule

Activity	Date	Responsibility
Requirement Analysis & Test Planning	Day 1	Vinita
Manual Test Case Design	Day 1	Vinita
Manual Test Execution & Bug Logging	Day 1	Vinita
Automation Framework Setup (POM + Cucumber)	Day 1	Palak
Automation Script Development	Day 1	Palak
Automation Test Execution	Day 2	Palak
Report Generation (Extent/Allure)	Day 2	Palak
Final Review & Test Summary Submission	Day 2	Vinita /Palak

➤ Test Deliverables

- Test Plan Document
- Manual Test Case Sheet
- Bug Report (in Jira or spreadsheet)
- Cucumber Feature Files
- Step Definition Classes
- Page Object Classes
- TestNG/JUnit Runner Files
- Execution Report (Extent or Allure)
- Final Summary Report

➤ Entry and Exit Criteria

Entry Criteria:

- Test environment is set up and accessible.
- Login page is functional.
- Required tools and dependencies are installed.
- Manual test cases are reviewed and approved.

Exit Criteria:

- All manual test cases executed with pass/fails status.
- All critical defects resolved and retested.
- Automation scripts show 95%+ pass rate.
- Test summary and reports submitted.

➤ Tools

Category	Tool
Test Automation	Selenium WebDriver
Programming	Java
BDD Framework	Cucumber
Build Tool	Maven
Test Execution	TestNG / JUnit
Reporting	Extent Reports / Allure
Manual Test Management	Google Sheets, Jira
IDE	IntelliJ IDEA / Eclipse

➤ Risks and Mitigation

Risk	Mitigation
Application downtime	Inform stakeholders, buffer time in schedule
Locators change frequently	Use robust locators and maintain POM
Incorrect test data	Keep valid/invalid data sets pre-defined
Report tool conflict	Use compatible version or switch between Extent

➤ Approvals

This project has been reviewed and approved by **Shiva Kumar MG (Head)**.

